Apprentissage Actif

Raphaël Roy, Yannis Bourree, Thomas Delpy, Clément Gicquel

1. Projet

a. Description

Les réseaux de neurones ont besoin d'apprendre à partir de nombreux exemples pour pouvoir être efficace. Des banques de données qui regroupent assez d'images pour l'entraînement des réseaux existent, cependant elles ne sont pas toutes égales en quantité. Les objets et animaux du quotidien sont bien représentés dans ces sets de données, mais les objets plus complexes le sont beaucoup moins. C'est le cas des images du domaine de la santé, comme des radios ou des échographies. Dans ces cas là, chaque image peut coûter très cher à labelliser, elles sont donc rares et entraîner un réseau à partir de ces échantillons devient compliqué.

C'est là qu'intervient l'apprentissage actif. Pour un réseau et un dataset donné, cela permet de sélectionner les données les plus pertinentes pour le réseau, c'est-à-dire celles qui vont améliorer au maximum les performances du réseau. Ainsi, en appliquant ces méthodes d'apprentissage actif, on réseau peut atteindre avec peu de données les résultats du même réseau avec beaucoup plus d'échantillons à disposition pour son apprentissage.

Le but de notre projet est de comparer les trois méthodes de sélection de données suivantes :

- Échantillonnage par marge
- Échantillonnage par moindre confiance
- Échantillonnage par entropie

Pour effectuer ces comparaisons, on teste les performances de 3 réseaux (VGGnet, ResNet, AlexNet) sur trois datasets (CIFAR10, CIFAR100, UrbanSound8K) en utilisant les différents méthodes de sélection des données à utiliser pour l'apprentissage.

b. Attentes

A l'aide de graphes de résultats, on veut pouvoir visualiser l'évolution des performances des réseaux neuronaux en fonction du nombre d'échantillons que l'on ajoute au dataset, c'est-à-dire le nombre d'exemples dont on va demander le label.

On s'attend à ce que la courbe de progression des performances de classification des réseaux augmentent significativement plus rapidement en sélectionnant méthodiquement les exemples plutôt qu'en les prenant au hasard.

Mis côte à côte, les graphes des résultats devraient nous permettre de distinguer quelle est la méthode de sélection la plus efficace pour un réseau et un dataset donné.

c. Limites

Nous aurions aimé tester l'apprentissage actif sur le dataset audio UrbanSound8K, mais par manque de temps nous n'avons utilisé que les dataset d'images CIFAR10 et CIFAR100.

Aussi nous n'avons pas réussi à utiliser les GPU de calcul Québec pour faire les expériences qui demandent énormément de temps. Nous avons donc utilisé nos CPU personnels, ce qui ne nous a pas permis de faire des expériences très longues et complètes.

2. Outils

a. Python

Le projet est développé en Python, un langage de programmation open source populaire dans la communauté de l'apprentissage machine et de l'apprentissage profond. Notamment, la bibliothèque Numpy nous permet de manipuler des tableaux à N-dimensionnels en Python. Aussi, la bibliothèque Scipy pour les modèles certaines fonctions appliquées aux statistiques, et matplotlib pour les représentations graphiques.
https://www.python.org

b. Pytorch

Les modeles de reseaux de neurones profond de notre projet sont développé à l'aide de PyTorch, une bibliothèque logicielle open source Python d'apprentissage machine qui s'appuie sur Torch. PyTorch est developpée par Facebook
PyTorch permet d'effectuer les calculs tensoriels nécessaires notamment pour l'apprentissage profond. Ces calculs sont optimisés, et effectués soit par le processeur (CPU), soit, lorsque c'est possible, par un processeur graphique (GPU) supportant CUDA.
https://pytorch.org

c. Github

Pour réaliser notre projet nous utilisons Github. Les systèmes de contrôle de version comme Git permettent aux utilisateurs de garder une trace de tous les changements qui se sont produits dans un projet. C'est une solution idéale pour travailler en collaboration sur des moyens à larges projets.

Git se concentre sur le paradigme de la ramification, où les changements peuvent être ajoutés et fusionnés de manière contrôlée, et logique. Chaque membre de l'équipe travaille sur sa propre version du code et peut proposer ou fusionner ses modifications lorsque de nouvelles fonctionnalités sont prêtes.

https://github.com

d. CookieCutter

Nous avons commencé par utiliser Cookie Cutter, un utilitaire de ligne de commande qui crée des projets à partir de modèles de projet. En effet, il existe un modèle permettant de créer une base très complète de paquet python, installable avec pip. Mais cela ne correspondait pas à notre projet qui n'avait pas pour but d'être un paquet python.
https://cookiecutter.readthedocs.io/en/1.7.0/#
https://github.com/cookiecutter/cookiecutter

3. Méthodes d'apprentissage actif

a. Aléatoire

La méthode aléatoire est la méthode par défaut qui n'utilise pas les principes d'apprentissage actif. Cette méthode choisit aléatoirement les échantillons à ajouter au dataset d'entraînement parmi tous les échantillons disponibles. On se sert des résultats obtenus via cette méthode comme référence de base pour nos réseaux de neurones. Le but de l'étude est donc de montrer qu'on peut atteindre, voire dépasser, les performances d'un apprentissage qui sélectionne aléatoirement les exemples utilisés pour l'entraînement versus un apprentissage les sélectionnant.

b. Échantillonnage par moindre confiance

Avec cette méthode, on sélectionne les exemples dont la probabilité de classification est la plus faible. C'est à dire que les exemples dont le réseau est le moins seront choisis.

Exemple : Soit A, B et C trois classes d'un dataset, et x1 et x2 deux échantillons du dataset.
Le réseau a classifié x1 et x2 avec les probabilités suivantes :

- x1 : A = 0.8, B = 0.1, C = 0.1
- x2 : A = 0.5, B = 0.3, C = 0.2

Le réseau a classifié les deux exemples comme appartenant à la classe A, mais dans le cas de x2, la probabilité est plus faible, c'est donc cet exemple qui sera sélectionné pour le prochain entraînement.

https://www.datacamp.com/community/tutorials/active-learning

c. Échantillonnage par marge

L'échantillonnage de marge est une méthode d'apprentissage active efficace qui calcule l'incertitude d'une instance x en comparant ses deux principales prédictions du modèle, max et $max2$. L'échantillonnage de marge ms est défini comme suit :
Pour un domaine cible D_T uniquement représenté par un ensemble de données non étiquetées ${D_T}^U$.

$$ms = sort_{\forall x \in {D_T}^U} (max(p(y_s)) - max2(p(y_s))$$

avec $max(p(y_s))$ et $max2(p(y_s))$ les deux meilleures probabilités d'appartenance de l'échantillon x, et $sort$ une fonction de tri.

Ainsi, cette fonction d'acquisition privilégie les échantillons qui minimisent la différence entre les deux premières prédictions des échantillons de la liste ms. (Bondu and Lemaire, 2020)

d. Échantillonnage par entropie

L'échantillonnage par entropie est une méthode d'apprentissage actif basée encore sur l'incertitude. On calcule l'entropie des probabilités de classification pour chaque image, et on garde les plus grandes:

$$ent = invsort_{\forall x \in {D_T}^U} (H(x))$$

avec $H(x)$ l'entropie des probabilités d'appartenance à chaque classe d'une image x, et $invsort$ une fonction de tri inversée (ordre décroissant).

4. Bases de données

Pour la réalisation de notre projet, nous avons choisi deux bases de données d'images, CIFAR-10 et CIFAR-100, et une d'extraits audio, UrbanSound8K.

a. CIFAR10

CIFAR-10 est une base de donnée qui comprend 60000 images couleur, de 32 pixels par 32 pixels, dans 10 classes, avec 6000 images par classe. Il y a 50000 images d'entraînement et 10000 images de test. Les dix classes sont:
Avion, voiture, oiseau, chat, cerf, chien, grenouille, cheval, bateau, camion

Exemple d'image issu de la base de donnée CIFAR-10: 

L'ensemble de données se divise en cinq lots d'entraînement, et un lot de test, contenant chacun 10 000 images.
Le lot de test contient exactement 1000 images sélectionnées au hasard dans chaque classe. Les lots d'entraînement contiennent les images restantes dans un ordre aléatoire, mais certains lots d'entraînement peuvent contenir plus d'images d'une classe que d'une autre. Entre eux, les lots de formation contiennent exactement 5000 images de chaque classe.

https://www.cs.toronto.edu/~kriz/cifar.html

b. CIFAR100

CIFAR-100 est une base de donnée identique à CIFAR-10, sauf qu'il comporte 100 classes contenant chacune 600 images. Il y a 500 images d'entraînement et 100 images de test par classe.
Les 100 classes du CIFAR-100 sont regroupées en 20 superclasses. Chaque image est accompagnée par :

- une étiquette "fine" associée à la classe à laquelle elle appartient
- une étiquette "grossière" associée à la superclasse à laquelle elle appartient.

Voici la liste des superclasses/classes de la CIFAR-100 :

mammifères aquatiques	castor, dauphin, loutre, phoque, baleine
poissons d'aquarium	poissons plats, raie, requin, truite
grands carnivores	ours, léopard, lion, tigre, loup
grands éléments extérieurs construits par l'homme	pont, château, maison, route, gratte-ciel
grandes scènes naturelles de plein air	nuage, forêt, montagne, plaine, mer
grands omnivores et herbivores	chameau, bovin, chimpanzé, éléphant, kangourou
mammifères de taille moyenne	renard, porc-épic, opossum, raton laveur, mouffette
invertébrés non insectes crabe	homard, escargot, araignée, ver
personnes	bébé, garçon, fille, homme, femme

reptiles	crocodile, dinosaure, lézard, serpent, tortue
petits mammifères	hamster, souris, lapin, musaraigne, écureuil
arbres	érable, chêne, palmier, pin, saule
véhicules 1	bicyclette, bus, moto, camionnette, train
véhicules 2	tondeuse à gazon, fusée, tramway, citerne, tracteur

https://www.cs.toronto.edu/~kriz/cifar.html

c. UrbanSound8K

UrbanSound8K est un ensemble de données contenant 8732 extraits sonores étiquetés (<=4s) de sons urbains provenant de 10 classes : air_conditioner, car_horn, children_playing, dog_bark, drilling, engine_idling, gun_shot, jackhammer, siren et street_music. Les classes sont tirées de la taxonomie des sons urbains.

Comme cité sur leur site, tous les extraits sont tirés d'enregistrements de terrain téléchargés sur www.freesound.org.
Les fichiers sont pré-triés en dix dossiers (dossiers nommés de fold1 à fold10) pour faciliter la reproduction et la comparaison avec les résultats de la classification automatique.

En plus des extraits sonores, un fichier CSV contenant des métadonnées sur chaque extrait est également fourni.

Les fichiers audio comprennent:

8732 fichiers audio de sons urbains (voir description ci-dessus) en format WAV. Le taux d'échantillonnage, la profondeur de bits et le nombre de canaux sont les mêmes que ceux du fichier original téléchargé sur Freesound.

Fichiers des métadonnées inclus: UrbanSound8k.csv

Ce fichier contient des métadonnées sur chaque fichier audio de l'ensemble de données.

https://urbansounddataset.weebly.com

https://pytorch.org/tutorials/beginner/audio_classifier_tutorial.html?highlight=audio

https://colab.research.google.com/github/pytorch/tutorials/blob/gh-pages/_downloads/audio_classifier_tutorial.ipynb#scrollTo=Dmp9W9MweeW8

5. Modèles de réseau de neurones convolutifs

a. VGG: Visual Geometry Group

VGG est un réseau neuronal convolutif (CNN) qui se caractérise par sa simplicité, n'utilisant que des couches convolutionnelles 3×3 empilées les unes sur les autres de plus en plus profondément.
La réduction de la taille du volume est assurée par une mise en commun maximale. Deux couches entièrement connectées, chacune comportant 4 096 nœuds, sont ensuite suivies par un classificateur softmax.
Il existe différentes versions tel que VGG16 ou VGG19, représentent le nombre de couches de poids dans le réseau (respectivement 16 ou 19).

http://www.robots.ox.ac.uk/~vgg/ (Simonyan and Zisserman, 2015)

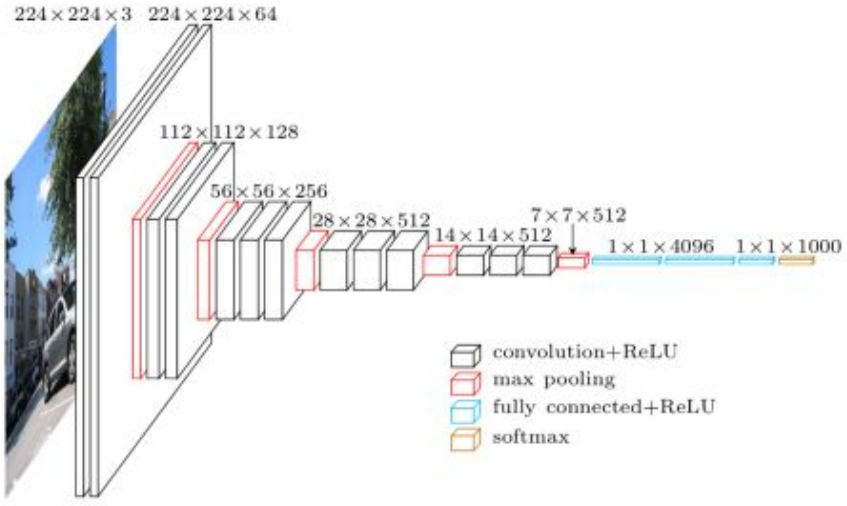


Figure: architecture du reseau VGG

b. ResNet: Residual Neural Network

Introduit par He et al. dans leur article de 2015, Deep Residual Learning for Image Recognition (He, Zhang, Ren and Sun, 2015), ResNet est un réseau s'appuyant sur des constructions connues des cellules pyramidales du cortex cérébral.

Ces réseaux neuronaux résiduels utilisent des connexions de saut (skip), ou des raccourcis pour sauter par-dessus certaines couches.
Contrairement aux architectures de réseau séquentielles traditionnelles, telles que VGG ou AlexNet, ResNet repose sur des modules de micro-architecture.

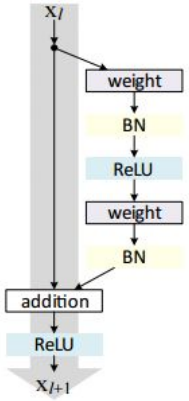


Figure: réseaux neuronaux résiduels

Le terme de micro-architecture fait référence à l'ensemble des "blocs de construction" utilisés pour construire le réseau. Un ensemble de blocs de construction de micro-architecture mène à la macro-architecture.
https://www.pyimagesearch.com/2017/03/20/imagenet-vggnet-resnet-inception-xception-keras/

c. AlexNet

AlexNet est un réseau neuronal convolutif (CNN), conçu par Alex Krizhevsky (Krizhevsky, Sutskever and Hinton, 2020), se composant de 5 couches convolutionnelles et de 3 couches entièrement connectées. AlexNet utilise la fonction d'activation ReLU non saturante, qui a montré une amélioration des performances d'entraînement par rapport au tanh et au sigmoïde.
https://www.learnopencv.com/understanding-alexnet/

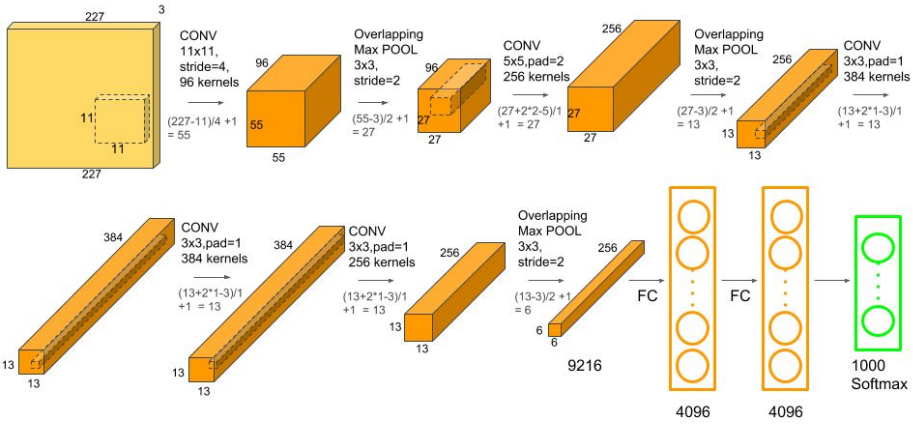


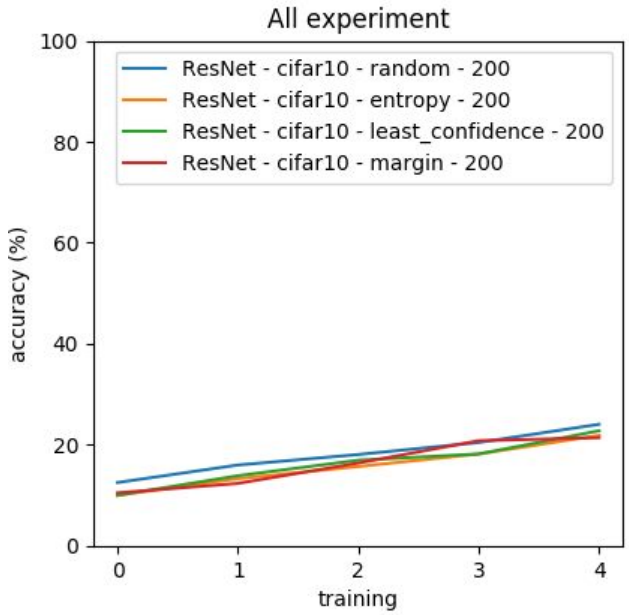
Figure: architecture du reseau AlexNet

6. Expérimentations

Passons maintenant à la phase d'expérimentation. Pour celle-ci nous avons choisi de lancer plusieurs expériences plutôt qu'une seule, car cela aurait pris bien plus de temps, surtout que nous les lancions sur nos ordinateurs dont la puissance de calcul est moindre, et limitée aux CPU.
Le but était d'étudier l'impact de l'apprentissage actif, dès lors chaque expérience faisait l'apprentissage actif avec les quatres méthodes vues précédemment.

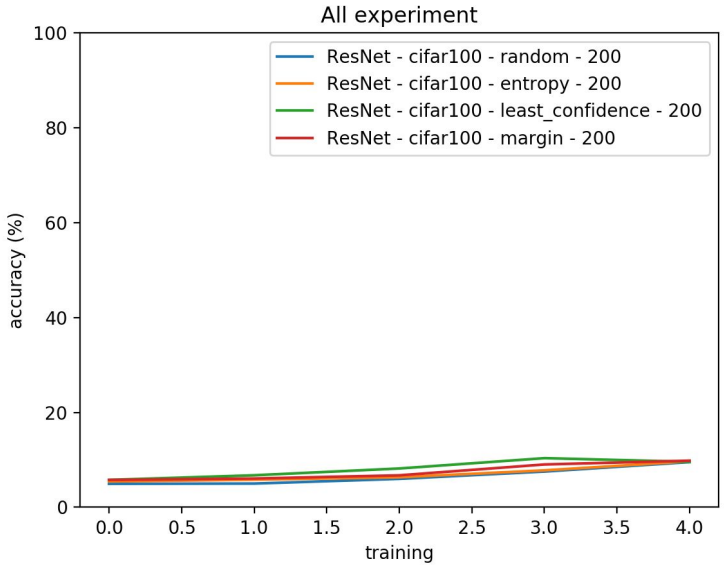
Les figures ci-dessous montrent l'évolution du pourcentage de bonne classification des images du set de test au cours des apprentissages actifs sur différents réseaux à convolution.
Nous avons choisi de faire cinq apprentissage par méthode, ce qui est déjà très long. C'est aussi pour cela que nous avons pris un seul k=200, c'est-à-dire que 200 images sont extraites de pool pour le premier entraînement, puis à la fin de celui-ci, pendant l'évaluation sur le set de validation, 200 images sont extraites de celui-ci et ajoutées au prochain set d'entraînement. Ce sont ces 200 images qui sont sélectionnés par la méthode d'apprentissage actif. Donc, pour résumer, le premier entraînement se fait sur 200 images, le deuxième sur 400, etc., jusqu'au cinquième qui en utilise 1000.

Voici ce que l'on obtient sur le réseau à convolution ResNet, sur le dataset CIFAR10, sachant que nous avions choisi 10 répétitions par entraînement - soit 10 *epochs*.



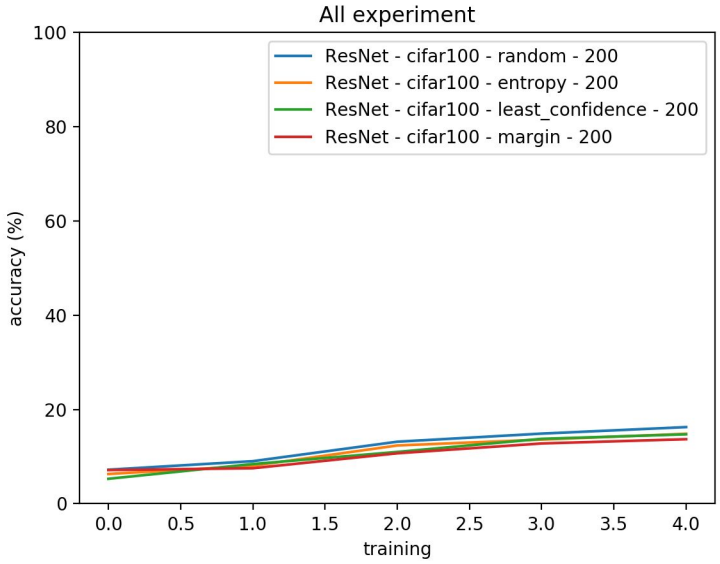
Nous pouvons voir qu'il y a un apprentissage car le pourcentage de bonne classification des images de test augmente au fur et à mesure des boucles d'apprentissage actif. Cependant, aucune méthode ne semble donner de meilleurs résultats que la sélection aléatoire.

Dans la figure suivante, l'expérience utilise le dataset CIFAR100. Il est important de noter que nous n'utilisons pas les 100 classes mais les 20 classes plus génériques. Dès lors, le réseau doit apprendre à reconnaitre 20 classes et non 10 comme précédemment.



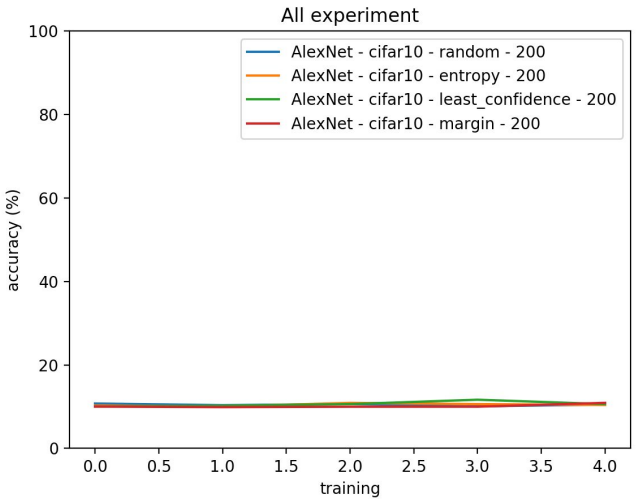
Nous pouvons toujours observer un apprentissage, mais très léger. Concernant les méthodes de sélection, celle de moindre confiance donne un meilleur résultat au troisième apprentissage mais finit par arriver au même pourcentage de réussite que les autres.

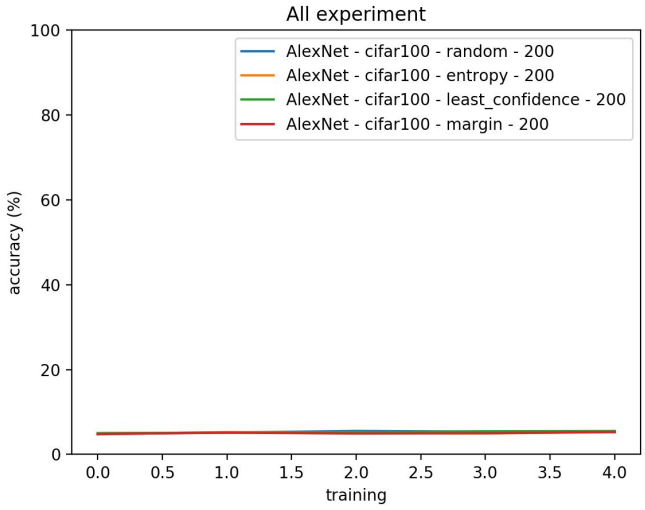
Nous avons lancé la même expérience, mais avec 30 *epochs* cette fois, voici le résultat.



Nous pouvons voir que le pourcentage de bonne classification est meilleur, car le réseau a vu plus de fois les mêmes images. Malheureusement, la méthode de sélection aléatoire donne un léger meilleur taux de classification.

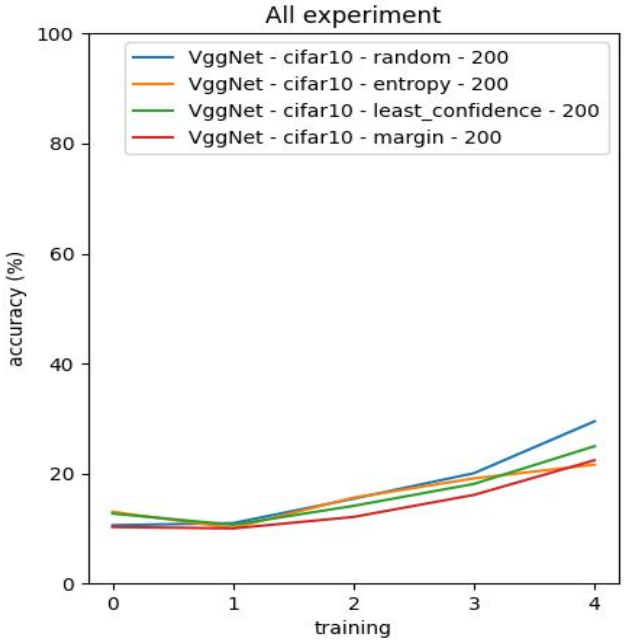
Passons maintenant au réseau à convolution AlexNet. Dans les expériences ci-dessous nous avons utilisé CIFAR10 et CIFAR100, et 10 *epochs* par entraînement.





Comme vous pouvez le voir, il n'y a pas d'apprentissage car le taux de classification stagne à 10% (1 chance sur 10) ou 5% (1 chance sur 20). Les méthodes de sélection sont alors indifférenciables à ce stade.

Pour finir, voici ce que l'on obtient avec le réseau VggNet sur le dataset CIFAR10 et 10 *epochs* par entraînement.



Nous pouvons observer un taux d'apprentissage qui augmente bien, jusqu'à environ 30%. Cependant c'est la méthode de sélection aléatoire qu détient ce score. Celle de moindre

confiance vient juste après, mais nous nous attendions à ce que le taux de classification soit meilleur pour les méthodes de sélection d'apprentissage actif.

7. Conclusion

Un des points limitants des expériences a été les méthodes de calcul utilisées pour déterminer les échantillons à "labelliser", soit dans notre cas ceux à ajouter au dataset d'entraînement. Pour trier les échantillons en fonction de leur pertinence pour le réseau, on utilise les fonctions d'évaluations. Dans notre projet, les résultats de ces évaluations étaient souvent très proches entre les échantillons, parfois même égaux. Cette imprécision constitue un facteur négatif pour l'apprentissage du réseau. En effet, à partir du moment où les fonctions d'évaluations donnent des résultats proches pour des échantillons qui se ressemblent, les k-exemples sélectionnés pour l'apprentissage actif ne sont dès lors plus des échantillons qui maximisent les capacités de classification des réseaux, mais des échantillons qui se ressemblent, et cela va donc diminuer l'apprentissage du réseau.

Les résultats obtenus sont donc en contradiction avec ceux qui étaient attendus, principalement pour la raison énoncée dans le paragraphe précédent. Pour améliorer les résultats, il faudrait faire tourner le programme plus longtemps, sur des datasets plus étendus pour mieux faire apparaître les différences. Avec suffisamment d'*epochs*, l'impact de l'apprentissage actif devrait se faire beaucoup plus sentir.

Enfin le deuxième point à améliorer est le choix des méthodes de sélection utilisées. Nos trois méthodes sont implémentables aiséments, mais ne fournissent pas une excellente précision. D'autres méthodes, comme l'arbre sur l'espace des versions, la réduction de l'erreur de généralisation du modèle ou l'échantillonnage par comité de modèles, sont censées donner une bien meilleure précision, mais nous n'avons pas su les comprendre correctement pour les implémenter.

Pour finir, nous sommes quand même très satisfaits de notre travail. Nous pensons avoir produit un code propre, compréhensible, réutilisable et modifiable. Ainsi, si l'occasion se présente, nous, ou quelqu'un d'autre tombant sur le dépôt GitHub, pourrions nous resservir de ce programme. Notre projet peut servir de base modulable pour un projet voulant intégrer l'apprentissage actif.

De plus, ce programme peut prendre un grand nombre d'arguments afin d'étudier l'apprentissage actif, et si l'équipement le permet, de faire des expériences très complètes. Car l'apprentissage actif reste quelque chose de long, fait de plusieurs apprentissages successifs.

8. Références

Apprentissage actif:
Aggarwal, U., Popescu, A. and Hudelot, C., 2020. *Active Learning For Imbalanced Datasets*. [online] Openaccess.thecvf.com. Available at: <http://openaccess.thecvf.com/content_WACV_2020/papers/Aggarwal_Active_Learning_for_Imbalanced_Datasets_WACV_2020_paper.pdf> [Accessed 9 April 2020].

Apprentissage actif:
Bondu, A. and Lemaire, V., 2020. *État De L'Art Sur Les Méthodes Statistiques D'Apprentissage*. [online] Vincentlemaire-labs.fr. Available at: <http://vincentlemaire-labs.fr/publis/rnti_2007_submited.pdf> [Accessed 9 April 2020].

AlexNet:
Krizhevsky, A., Sutskever, I. and Hinton, G., 2020. *Imagenet Classification With Deep Convolutional Neural Networks*. [online] Papers.nips.cc. Available at: <https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf> [Accessed 11 April 2020].

VGG:
Simonyan, K. and Zisserman, A., 2015. *VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION*. [online] Arxiv.org. Available at: <https://arxiv.org/pdf/1409.1556.pdf> [Accessed 11 April 2020].

ResNet:
He, K., Zhang, X., Ren, S. and Sun, J., 2015. *Deep Residual Learning For Image Recognition*. [online] Arxiv.org. Available at: <https://arxiv.org/pdf/1512.03385.pdf> [Accessed 11 April 2020].

Généralités sur l'apprentissage actif:
Hosein, S., 2018. *Active Learning: Curious AI Algorithms*. [online] Datacamp. Available at: <https://www.datacamp.com/community/tutorials/active-learning> [Accessed 11 April 2020].